POWER SAVINGS IN DYNAMIC REGISTER FILE STRUCTURES

Arjun P. Chandran

Gregg K. Tsujimoto

5   Anup S. Mehta

FIELD OF THE INVENTION

10   The present invention relates generally to
microprocessors and, more particularly, to Retirement
Payload Arrays used in the Retirement Window of
Instruction Scheduling Units of microprocessors.

15

BACKGROUND OF THE INVENTION

    With the emergence of an electronics market that
stresses portability, compact size, lightweight and the
20 capability for prolonged remote operation, a demand has
arisen for low power circuits and systems. This demand
has motivated circuit designers to depart from
conventional circuit designs and venture into more
power and space efficient alternatives. Nowhere is the
25 minimization of power and space usage more critical
than in the processors employed in computer systems.

    Processor architectures can be represented as a
collection of interacting functional units as shown in

FIG.1. These functional units, discussed in greater

detail below, perform the functions of fetching

instructions and data from memory, decoding fetched

instructions, scheduling instructions to be executed,

5   executing the instructions, managing memory

transactions, retiring instructions and interfacing

with external circuitry and devices.

The present invention is described in terms of

apparatus and methods particularly useful in a highly

10   pipelined and superscalar processor 102 shown in block

diagram form in FIG.1 and FIG.2. The particular

examples represent implementations that can be used to

issue and execute multiple Instructions Per Cycle (IPC)

and are amenable to high clock frequency operations.

15   However, it is expressly understood that the inventive

features of the present invention may be usefully

embodied in a number of alternative processor

architectures that will benefit from the performance

features of the present invention. Accordingly, these

20   alternative embodiments are equivalent to the

particular embodiments shown and described herein.

FIG.1 shows a typical general-purpose computer

system 100 incorporating a processor 102 in accordance

with the present invention. Computer system 100

25   comprises an address/data bus 101 for communicating

information, processor 102 coupled with bus 101 through

input/output (I/O) interface 103 for processing data

and executing instructions, and memory system 104

-2-

coupled with bus 101 for storing information and instructions for processor 102. Memory system 104 comprises, for example, cache memory 105 and main memory 107. Cache memory 105 can include one or more

5   levels of cache memory. In a typical embodiment, processor 102, I/O interface 103, and some or all of cache memory 105 may be integrated in a single integrated circuit, although the specific components and integration density are a matter of design choice

10 selected to meet the needs of a particular application.

User I/O devices 106 are coupled to bus 101 and are operative to communicate information in appropriately structured form to and from the other parts of computer 100. User I/O devices may include a

15 keyboard, mouse, card reader, magnetic or paper tape, magnetic disk, optical disk, or other available devices, including another computer. Mass storage device 117 is coupled to bus 101, and may be implemented using one or more magnetic hard disks,

20 magnetic tapes, CDROMs, large banks of random access memory, or the like. Mass storage 117 may include computer programs and data stored therein. Some or all of mass storage 117 may be configured to be incorporated as a part of memory system 104.

25 In a typical computer system 100, processor 102, I/O interface 103, memory system 104, and mass storage device 117, are coupled to bus 101 formed on a printed circuit board and integrated into a single housing as

suggested by the dashed-line box 108. However, the particular components chosen to be integrated into a single housing is based upon market and design choices.

Display device 109 is used to display messages,
5   data, a graphical or command line user interface, or other communications with the user. Display device 109 may be implemented, for example, by a cathode ray tube (CRT) monitor, liquid crystal display (LCD) or any available equivalent.

10   FIG.2 illustrates principle components of processor 102 in greater detail in block diagram form. It is contemplated that processor 102 may be implemented with more or fewer functional components and still benefit from the apparatus and methods of the
15   present invention unless expressly specified herein. In addition, functional units are identified using a precise nomenclature for ease of description and understanding, but other nomenclature often is used to identify equivalent functional units.

20   Instruction fetch unit (IFU) 202 comprises instruction fetch mechanisms and includes, among other things, an instruction cache for storing instructions, branch prediction logic, and address logic for addressing selected instructions in the instruction
25   cache. The instruction cache (I$) is commonly a portion of the level one cache (L1$), with another portion of the L1 cache dedicated to data storage (D$). IFU 202 fetches one or more instructions at a time by

appropriately addressing the instruction cache. The instruction cache feeds addressed instructions to instruction rename unit (IRU) 204. Typically, IFU 202 fetches multiple instructions each cycle, and in a

5   specific example fetches eight instructions each cycle.

In the absence of a conditional branch instruction, IFU 202 addresses the instruction cache sequentially. The branch prediction logic in IFU 202 handles branch instructions, including unconditional

10   branches. An outcome tree of each branch instruction is formed using any of a variety of available branch prediction algorithms and mechanisms. More than one branch can be predicted simultaneously by supplying sufficient branch prediction resources. After the

15   branches are predicted, the address of the predicted branch is applied to the instruction cache rather than the next sequential address.

IRU 204 comprises one or more pipeline stages that include instruction renaming and dependency checking

20   mechanisms. The instruction renaming mechanism is operative to map register specifiers in the instructions to physical register locations and to perform register renaming to minimize dependencies. IRU 204 further comprises dependency checking

25   mechanisms that analyze the instructions fetched by IFU 202 amongst themselves, and against those instructions installed in ISU 206, to establish true dependencies.

IRU 204 outputs renamed instructions to instruction
scheduling unit (ISU) 206.

ISU 206 receives renamed instructions from IRU 204
and registers them for execution. Upon registration,
5   instructions are deemed "live instructions" in a
specific example. ISU 206 is operative to schedule and
dispatch instructions as soon as their dependencies
have been satisfied into an appropriate execution unit
(e.g., integer execution unit (IEU) 208, or floating
10   point and graphics unit (FGU) 210). ISU 206 also
maintains trap status of live instructions. ISU 206
may perform other functions such as maintaining the
correct architectural state of processor 102, including
state maintenance when out-of-order instruction issue
15   logic is used. ISU 206 may include mechanisms to
redirect execution appropriately when traps or
interrupts occur and to ensure efficient execution of
multiple threads where multiple threaded operation is
used.

20   ISU 206 also operates to retire executed
instructions when completed by IEU 208 and FGU 210.
ISU 206 performs the appropriate updates to
architectural register files and condition code
registers upon complete execution of an instruction.
25   ISU 206 is responsive to exception conditions and
discards or flushes operations being performed on
instructions subsequent to an instruction generating an
exception in the program order. ISU 206 quickly

removes instructions from a mispredicted branch path and initiates IFU 202 to fetch from the correct branch address. An instruction is retired when it has finished execution and all older instructions have

5 retired. Upon retirement, the instruction's result is written into the appropriate register file and it is no longer deemed a "live instruction".

IEU 208 includes one or more pipelines, each pipeline comprising one or more stages that implement

10 integer instructions. IEU 208 also includes mechanisms for holding the results and state of speculatively executed integer instructions. IEU 208 functions to perform final decoding of integer instructions before they are executed on the execution units and to

15 determine operand bypassing amongst instructions concurrently in execution on the processor pipelines. IEU 208 executes all integer instructions including determining correct virtual addresses for load/store instructions. IEU 208 also maintains correct

20 architectural register state for a plurality of integer registers in processor 102.

FGU 210 includes one or more pipelines, each comprising one or more stages that implement floating point instructions. FGU 210 also includes mechanisms

25 for holding the results and state of speculatively executed floating point and graphics instructions. FGU 210 functions to perform final decoding of floating point instructions before they are executed on the

execution units and to determine operand bypassing

amongst instructions concurrently in execution on the

processor pipelines. In the specific example, FGU 210

includes one or more pipelines dedicated to

5   implementing special purpose multimedia and graphics

instructions that are extensions to standard

architectural instructions for a processor. FGU 210

may be equivalently substituted with a floating point

unit (FPU) in designs in which special purpose graphics

10   and multimedia instructions are not used. FGU 210

preferably includes mechanisms to access single and/or

double precision architectural registers as well as

single and/or double precision rename registers.

A data cache memory unit (DCU) 212 shown in Fig.

15   2, including cache memory 105 shown in FIG.1, functions

to buffer memory reads from off-chip memory through

external interface unit (EIU) 214. Optionally, DCU 212

also buffers memory write transactions. DCU 212

comprises two hierarchical levels of cache memory on-

20   chip (L1$ and L2$) and a third cache level (L3$)

accessible through EIU 214. DCU 212, alternatively

referred to as the data cache subsystem, comprises

separate instruction and data caches (I$ and D$) at the

primary level 1 cache L1$, a unified on-chip level 2

25   cache L2$ and a unified external level 3 cache L3$.

DCU 212 also includes controller logic and associated

queues at each level. One or more of the cache levels

within DCU 212 may be read only memory to eliminate the logic associated with cache writes.

FIG.3 is a high-level block diagram showing the fundamental components of ISU 206 from FIG.2. As discussed above, ISU 206 receives renamed instructions from IRU 204 (FIG.2) and registers them for execution. This function of ISU 206 is performed by scheduling window 301 (FIG.3). In addition, as also discussed above, ISU 206 operates to retire executed instructions when completed by IEU 208 and FGU 210 (FIG.2). This function is performed, in large part, by retirement window 303 of ISU 206 (FIG.3). Retirement window 303 typically includes Instruction Retirement Logic 305, hereinafter referred to as IRL 305, and Retirement Payload Array 307, hereinafter referred to as RPA 307.

One of the functions of IRL 305 is to generate two important signals, or vectors, a retire pointer signal, hereinafter referred to as signal "READ POINTER" and an advance pointer signal, hereinafter referred to as signal "ADVANCE POINTER". The signals "READ POINTER" and "ADVANCE POINTER" are coupled from IRL 305 to RPA 307 by lines 309 and 311, respectively, in FIG.3. The signals "READ POINTER" and "ADVANCE POINTER" are also discussed in more detail below.

FIG.4 shows one example of a typical RPA 307. As seen in FIG.4, RPA 307 is comprised of M-rows, R0 to RM, and N-columns, C0 to CN, of memory cells, such as exemplary memory cells 405 and 406. As shown in FIG.4,

each memory cell, such as exemplary memory cells 405 and 406, is coupled to a read word line (RWL), such as read word lines RWL0 to RWLM, and each memory cell in a given row R0 to RM is coupled to the same RWL, RWL0 to

5 RWLM, respectively. As also shown in FIG.4, each memory cell, such as exemplary memory cells 405 and 406, is coupled to a read bit line (RBL), such as read bit lines RBL0 to RBLN, and each memory cell in a given column C0 to CN is coupled to the same RBL, RBL0 to

10 RBLN, respectively.

As also shown in FIG.4, each RBL, RBL0 to RBLN, is coupled to a corresponding pre-charge device, PC0 to PCN, respectively, and a sensing device, S0 to SN, respectively. Consequently: RBL0 is coupled to PC0 and

15 S0; RBL1 is coupled to PC1 and S1; RBL2 is coupled to PC2 and S2; RBL3 is coupled to PC3 and S3; RBLN-3 is coupled to PCN-3 and SN-3; RBLN-2 is coupled to PCN-2 and SN-2; RBLN-1 is coupled to PCN-1 and SN-1; RBLN is coupled to PCN and SN. Pre-charging and pre-charge

20 devices, such as PC0 to PCN, are well known in the art. Pre-charge devices PC0 to PCN typically consist of various well-known elements or structures such as P-FETs, NFETs and the like. In addition, RBL sensing, and sensing devices, such as S0 to SN, are also well

25 known in the art and sensing devices S0 to SN typically consist of various well-known elements or structures such as latches, cross coupled latches and the like.

RPA 307 also includes read pointer 450. The signal "READ POINTER" (not shown) from IRL 305 is used to generate read pointer 405. In addition, the signal "ADVANCE POINTER" (not shown) from IRL 305 is used to

5 advance read pointer 450 from one read word line, such as read word line RWL2 in FIG.4, to a new read word line, such as read word line RWL3 in FIG.4, as shown by the dotted line pointer 450A in FIG.4. The signal "ADVANCE POINTER" from IRL 305 is active only when read

10 pointer 450 is moved or shifted. In all other instances, signal "ADVANCE POINTER" is inactive.

One embodiment of RPA 307 is a 192 column, 16-read word line register file structure employing a dynamic, full swing pull down read mechanism. Consequently, in

15 this one embodiment of a RPA 307, N is equal to 192 and M is equal to 16.

In prior art operation of RPA 307, pre-charge devices PC0 to PCN pre-charged RBL0 to RBLN, respectively, on each "A" phase, i.e., the pre-charge

20 phase, of a clock signal. Then on each "B" phase, i.e., the evaluate phase, of the clock signal, read pointer 450 indicated which read word line, RWL0 to RWLM, was to be read. Sensing devices S0 to SN would then sense their respective RBLs, RBL0 to RBLN, to

25 yield an "updated" result. Consequently, in the prior art, RPA 307 would read RPA 307 each time the primary clock switched to the read or "B" phase, regardless of whether read pointer 450 had advanced. In the prior

-11-

art, the evaluated "new" result was then sampled by the free running sensing devices S0 to SN and the evaluated result was then typically latched until the next "B" phase of the clock signal initiated a new read.

5   In the prior art method described above, if read pointer 450 did not shift, i.e., advance or move read word lines, RPA 307, and sensing devices S0 to SN, continued to read and "update" data, and dissipate significant read power, even if the read data was the
10   same as that of the previous "B" phase. Since, it often was the case that the read data was the same as that of the previous "B" phase, using the prior art mechanisms, Read Bit Lines (RBLs), RBL0 to RBLN would discharge every "B" phase even when the data in the
15   array was the same for multiple cycles.

Thus, by way of example, using prior art methods with the embodiment of an RPA 307 that is a 192 column, 16 row register file structure employing a dynamic, full swing pull down read mechanism discussed above, in
20   each "B" phase of the primary clock, all 192 bit lines, RBL0 to RBL192 (not shown) of the RPA 307 would potentially discharge. This resulted in a significant waste of power as all 192 RBLs were read and power was used to rewrite identical data repeatedly with each
25   shift of the system clock to the "B" phase.

FIG.5 shows a portion of exemplary prior art column C1 of RPA 307 of FIG.4. While prior art column C1 is chosen for exemplary purposes, the structure of

-12-

FIG.5, and the following discussion, is equally
applicable to any of the prior art columns C1 to CN
shown in FIG.4. As seen in FIG.5, prior art column C1
included pre-charge device PC1, in this case a PFET,
and sensing device S1. Shown in FIG.5 are rows R0, R1
and RM. In the example of FIG.5, each row R0 to RM
includes a corresponding pull-down device PD0 to PDN,
respectively. In actual practice, as shown in FIG.4,
each row R0 to RM would include numerous memory cells
equal to the number of columns "N". In the prior art,
each pull-down device PD0 to PDM was coupled to RBL1 as
shown.

In the prior art, each pull-down device PD0 to PDM
was coupled to the output 517, 519, and 521,
respectively, of a corresponding one of Nor-Gates 501,
503, and 505, respectively. A first input line 500,
504, and 507 of Nor-Gates 501, 503, and 505,
respectively, was coupled to receive signals CELL0_1,
CELL1_1, CELLM_1, respectively. Signals CELL0_1,
CELL1_1, CELLM_1 were signals representing the contents
of corresponding memory cells, such as exemplary memory
cells 405 and 406 in FIG.4. In the prior art, second
input lines 511, 513, and 515 of Nor-Gates 501, 503,
and 505, respectively, were coupled directly to the
signal "READ POINTER".

As discussed above with respect to FIG.4, In the
prior art, read pointer 450 (FIG.4) selected one of
memory cells (not shown) and the contents of the

-13-

selected memory cell were then made available on RBL1 (FIG.5) when the "B" phase of the signal CLK was begun, i.e., when CLK went low, the contents of the selected memory cell, was coupled to RBL1 and sensing device S1.

5 As also discussed above, using the prior art method and structure of FIG.5, if read pointer 450 (FIG, 4) did not shift, i.e., advance or move rows, prior art RPA 307 continued to read data on each "B" phase of CLK, and dissipate read power, even if the read data was the

10 same as that of the previous "B" phase. Consequently, a significant amount of power was wasted.

The waste of power associated with the prior art methods is further illustrated in FIG.5A. FIG.5A is a signal diagram for prior art column C1 showing signals:

15 CLK 550, the system clock; READ POINTER 551 from IRL 305 (FIG.3), ADVANCE POINTER 552 from IRL 305 (FIG.3) and READ 553, from output 517 of Nor-Gate 501, as an example. As shown in FIG.5A, the signal "READ" at output 517 of Nor-Gate 501, for example, is active, and

20 a read of RPA 307 (FIG.4) is initiated, in each "B" phase of signal CLK 550. That is to say, at every time signal CLK 550 is in the "B" phase, i.e., at times T2, T4 and T6, signal READ 553 is active. However, as indicated at point 555 of time T4, the signal ADVANCE

25 POINTER 552 is active only at time T4. Consequently, only at time T4 has the read pointer 450 (FIG.4) actually advanced. Therefore, only at point 555 of time T4 has the data of RPA 307 (FIG.4) changed. As a

-14-

result, the reads initiated at times T2 and T6 by signal READ 553 result in reading and rewriting the same data from the previous cycle. Clearly, this is a waste of energy.

5   What is needed is a method and apparatus for controlling when data from a RPA is read so that reads occur only when there is new data to be read.

10 **SUMMARY OF THE INVENTION**

       According to the present invention, the pointer advance signal "ADVANCE POINTER" from the Instruction Retirement Logic (IRL) of the Instruction Scheduling

15 Unit (ISU) is utilized to provide conditional read signals. Consequently, according to the invention, a read of the RPA is completed only if it is determined that the read word line being read in the current cycle is not the same read word line that was read in the

20 previous cycle. According to the invention, if the read word line is the same, the RPA read is cut off, i.e. the bitlines remain pre-charged, and no read power is dissipated reading the unchanged data.

       In contrast, as discussed above, in the prior art,

25 a read operation was initiated on the RPA every "B" phase of the clock signal regardless of whether the read pointer was in the same position as the previous cycle or not. Thus, in the prior art, in each "B"

phase of the clock, all read bit lines of each of the
columns of the RPA could discharge leading to wasteful
power dissipation.

Using the method and structure of the present
5   invention, the RPA read is activated only when the read
pointer shifts and there is new data to be read.
According to the invention, at all other times, i.e.,
when there is no change in the data, the RPA holds the
results of the previous read operation. Consequently,
10 using the method and structure of the invention, no
power is dissipated making repeated reads of the same
data.   Therefore, the method and structure of the
present invention is more efficient and the power
savings within the RPA translates into lower risk for
15 Joule Heating and electro-migration problems.

In addition, the method and structure of the
present invention takes advantage of the pointer
advance signal "ADVANCE POINTER" already being
generated by the IRL to determine when the read pointer
20 has shifted. Consequently, the present invention can
be readily adapted to existing architectures and
designs.

In addition, as discussed in more detail below,
one embodiment of the present invention includes the
25 addition of only minimal components. Consequently, the
method and structure of the invention has minimal
layout and Design for Test (DFT) implications.

-16-

It is to be understood that both the foregoing
general description and following detailed description
are intended only to exemplify and explain the
invention as claimed.

5

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated
10   in, and constitute a part of this specification,
illustrate embodiments of the invention and, together
with the description, serve to explain the advantages
and principles of the invention. In the drawings:

FIG.1 shows, in block diagram form, a computer
15   system in accordance with one embodiment of the present
invention;

FIG.2 shows a processor in block diagram form in
accordance with one embodiment of the present
invention;

20   FIG.3 is a high-level block diagram showing the
fundamental components of the Instruction Scheduling
Unit (ISU) from FIG.2;

FIG.4 shows one example of a typical Retirement
Payload Array (RPA) from FIG.3 in accordance with one
25   embodiment of the present invention;

FIG.5 shows a typical prior art column of a
Retirement Payload Array (RPA) from FIG.4;

FIG.5A is a signal diagram for prior art column C1 showing signals: CLK, the system clock; READ POINTER from the IRL, ADVANCE POINTER from the IRL and READ;

FIG.6 is a flow chart of one embodiment of the method of the present invention;

FIG.7 shows a portion of an exemplary column C1 of an RPA which has been modified in accordance with the principles of one embodiment of the present invention; and

FIG.8 is a signal diagram for one embodiment of a modified column C1 in accordance with the principles of the invention showing signals: CLK, the system clock; READ POINTER, generated by the IRL, ADVANCE POINTER, generated by the IRL and CONDITIONAL READ generated in accordance with the invention.

DETAILED DESCRIPTION

The invention will now be described in reference to the accompanying drawings. The same reference numbers may be used throughout the drawings and the following description to refer to the same or like parts.

According to the present invention, the pointer advance signal "ADVANCE POINTER" (852 in FIG.8) from the Instruction Retirement Logic Block (305 in FIG.3) of the Instruction Scheduling Unit (206 in FIG.s 2 and

-18-

3) is utilized to provide conditional read signal
"CONDITIONAL READ" (853 in FIG.8) and a read of the RPA
(307 in FIG.3 and 4) is completed only if it is
determined that the read word line (RWL0 to RWLM in
FIG.4) being read in the current cycle is not the same
read word line that was read in the previous cycle.
According to the invention, if the read word line is
the same, the RPA read is cut off, i.e. the bitlines
remain pre-charged, and no read power is dissipated
reading the unchanged data.

In contrast, as discussed above, in the prior art,
a read operation was initiated on the RPA every "B"
phase (T2, T4 and T6 in FIG.s 5A and 8) of the clock
signal (CLK 550 in FIG.5A and 850 in FIG.8) regardless
of whether the read pointer (450 in FIG.4) was in the
same position as the previous cycle or not. Thus, in
the prior art, in each "B" phase of the clock, all read
bit lines (RBL0 to RBLN in FIG.4) of the RPA could
discharge leading to wasteful power dissipation.

Using the method and structure of the present
invention, the RPA read is activated only when the read
pointer shifts and there is new data to be read.
According to the invention, at all other times, i.e.,
when there is no change in the data, the RPA holds the
results of the previous read operation. Consequently,
using the method and structure of the invention, no
power is dissipated making repeated reads of the same
data. Therefore, the method and structure of the

-19-

present invention is more efficient and runs cooler. This power savings within the RPA translates into lower risk for Joule Heating and electro-migration problems.

In addition, the method and structure of the
5 present invention takes advantage of the pointer advance signal "ADVANCE POINTER" already being generated by the IRL to determine when the retire pointer signal "READ POINTER" (552 in FIG.5A and 852 in FIG.8) has shifted. Consequently, the present
10 invention can be readily adapted to existing architectures and designs.

In addition, as discussed in more detail below, one embodiment of the present invention has minimal layout and Design for Test (DFT) implications.
15 It is to be understood that both the foregoing general description and following detailed description are intended only to exemplify and explain the invention as claimed.

FIG.6 is a flow chart of one embodiment the
20 method of the present invention. FIG.8 is a signal diagram for one embodiment of the invention showing signals: CLK 850, the system clock; READ POINTER 851, generated by the IRL, ADVANCE POINTER 852, generated by the IRL and CONDITIONAL READ 853 generated in
25 accordance with the invention. The method of one embodiment of the invention will now be described with reference to FIG.s 6 and 8.

-20-

At Start 600, a new cycle of the clock signal (CLK 850 in FIG.8) begins. At 601 in FIG.6, a determination is made as to whether the read pointer has selected the read word line in question. If, as at 603, the answer

5   is "NO" the read pointer has not selected the read word line in question, then this read word line is not selected and no read of the read word line in question will take place as shown in 604. If, as at 605, the answer is "YES" and the read pointer has selected the

10   read word line in question, then the method moves on to 607.

At 607, a determination is made as to whether the clock signal (CLK 850 in FIG.8) has shifted to the "evaluate" or "B" phase (T2, T4 and T6 in FIG.8). If,

15   as at 609 in FIG.6, the answer is "NO" the clock signal has not shifted to the evaluate phase, then no read of the read word line in question will take place and the method must, as shown in 611, wait for a shift of the clock signal to the evaluate phase. If, as at 613, the

20   answer is "YES", the read pointer has selected the read word line in question and the clock has shifted to the evaluate phase, then the method moves on to 615.

At 615, a determination is made as to whether the read pointer has moved since the last read cycle. As

25   discussed in more detail below, in one embodiment of the invention, this determination is made by receiving the signal "ADVANCE POINTER" (852 in FIG.8) and checking to see if this signal has gone active,

-21-

indicating a read pointer movement (see point 855 in FIG.8). If, as at 617 in FIG. 6, the answer is "NO" the read pointer has not moved since the last read cycle (see times T1, T2, T3, T5 and T6 in FIG.8), then a read is not performed as shown by 619 in FIG.6 and the method ends at 621 until the next clock cycle. If, as at 623, the answer is "YES", the read pointer has selected the read word line in question, the clock has shifted to the evaluate phase and the read pointer has moved since the last read cycle (see time T4 in FIG.8), then signal CONDITIONAL READ 853 goes active (see point 857 in FIG.8) and an RPA read is performed at 625 in FIG.6. The data is then held at 627 until the next movement of the read pointer is combined with a shift of the clock signal to the evaluate phase.

FIG.7 shows a portion of exemplary modified column C1 of an RPA, such as RPA 307 of FIG.4, which has been modified in accordance with the present invention. While modified column C1 is chosen for exemplary purposes, the structure of FIG.7, and the following discussion, is equally applicable to any of the columns C1 to CN shown in FIG.4 that are modified according to the present invention.

As seen in FIG.7, modified column C1 includes pre-charge device PC1, in this case a PFET, and sensing device S1. Also shown in FIG.7 are rows R0, R1 and RM. In the embodiment of the invention shown in FIG.7, each row R0 to RM includes a corresponding pull-down device

-22-

PD0 to PDM, respectively. In actual practice, as shown in FIG.4, each row R0 to RM would include numerous memory cells (not shown) and numerous pull-down devices PD0 to PDM equal to the number of columns "N". As shown in FIG.7, in one embodiment of the invention, each pull-down device PD0 to PDM is coupled to a read bit line, RBL1, as shown.

According to one embodiment of the invention, each pull-down device PD0 to PDM is coupled to the output 717, 719, 721, respectively, of a corresponding one of Nor-Gates 701, 703, and 705, respectively. According to the invention, a signal "CONDITIONAL READ" (853 in FIG.8) is generated at outputs 717, 719, and 721, respectively, of a corresponding one of Nor-Gates 701, 703, and 705 (FIG.7), respectively. When active, the signal "CONDITIONAL READ" initiates a read of RPA 307 (FIG.4).

As seen in FIG.7, a first input line 700, 704, and 707 of Nor-Gates 701, 703, and 705, respectively, is coupled to receive signals CELL0_1, CELL1_1, CELLM_1, respectively. Signals CELL0_1, CELL1_1, CELLM_1 are signals representing the contents of corresponding memory cells, such as exemplary memory cells 405 and 406 in FIG.4. Second input lines 711, 713, and 715 of Nor-Gates 701, 703, 705, respectively, are coupled to the outputs 724, 726, and 728 of conditional read circuits CRC0, CRC1 and CRCM, respectively.

-23-

In one embodiment of the invention, conditional read circuits CRC0, CRC1 and CRCM each include a NAND-Gate, 731, 733 and 735, respectively, and an inverter, 751, 753 and 757, respectively. In one embodiment of

5   the invention the signal CLK, i.e., the signal "CLK" is coupled to input lines 761, 763 and 767 of inverters 751, 753 and 757, respectively. The inverted "CLK" signal is then coupled to the first input lines 743, 745 and 747 of NAND-Gates, 731, 733 and 735,

10   respectively. In one embodiment of the invention, the second input lines 737, 739, and 741 of NAND-Gates, 731, 733 and 735, respectively, are coupled to the signal ADVANCE POINTER (853 of FIG.8) from IRL 305 (FIG.3). As shown in FIG.7, in one embodiment of the

15   invention, the outputs 723, 725, and 727 of NAND-Gates, 731, 733 and 735, respectively, are coupled to outputs 724, 726, and 728 of conditional read circuits CRC0, CRC1 and CRCM, respectively.

FIG.8 is a signal diagram for modified column C1

20   of FIG.7 showing signals: CLK 850, the system clock; READ POINTER 851 from IRL 305 (FIG.3), ADVANCE POINTER 852 from IRL 305 (FIG.3) and CONDITIONAL READ 853, generated from output 717 of Nor-Gate 701 in FIG.7, as an example. As shown in FIG.8, the signal CONDITIONAL

25   READ 853 at output 717 of Nor-Gate 701, for example, is active, and a read of RPA 307 (FIG.4) is initiated, only when the signal ADVANCE POINTER 852 from IRL 305 (FIG.3) is active, indicating the read pointer (450 in

-24-

FIG.4) has advanced, and the "B" phase of signal CLK 850 is begun (see time T4 ion FIG.8). That is to say, at every other time when the signal CLK 850 is in the "B" phase and the signal ADVANCE POINTER 852 from IRL 305 (FIG.3) is not active, i.e., at times T1, T2, T3, T5 and T6, the signal CONDITIONAL READ 853 is not active and a read of the RPA is not initiated. However, as indicated at point 855 of time T4, when the signal ADVANCE POINTER 852 is active, and the signal CLK 850 is in the "B" phase, signal CONDITIONAL READ 853 is active and a read of the RPA is initiated. Consequently, according to the invention, a read of the RPA is initiated only when the read pointer has advanced and there is new data to read. At all other times no read is initiated and power is saved.

In contrast, recall that in the prior art, as shown in FIG.s 5 and 5A and discussed above, second input lines 511, 513, and 515 of Nor-Gates 501, 503, 505, respectively, were coupled directly to the signal "CLK". Therefore, even if read pointer 450 (FIG, 4) did not shift, i.e., advance or move rows, and there was no new data, signal READ 553 (FIG.5A) still went active with each "B" phase of CLK 550 and a read was initiated. Consequently, in the prior art, RPA 307 (FIG.4) continued to read data and dissipate read power even if the read data was the same as that of the previous "B" phase and a significant amount of power was wasted.

-25-

According to the invention, the pointer advance signal "ADVANCE POINTER" is logically combined with the primary clock signal "CLK" and the read pointer signal "READ POINTER" to compute whether the read pointer has advanced since the last read. If the read pointer has advanced, then there is new data to read and a read of the RPA is initiated, otherwise the read is disabled. Consequently, according to the invention, a read of the RPA occurs only when there is new data to be read and the retire pointer shifts in the B phase of the clock.

The foregoing description of an implementation of the invention has been presented for purposes of illustration and description only, and therefore is not exhaustive and does not limit the invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practicing the invention.

For example, for illustrative purposes, specific embodiments of the invention were shown with specific Conditional Read Circuits, CRC0 to CRCM, and with specific gates. However, those of skill in the art will recognize that different gates and combination of gates could be used to form Conditional Read Circuits, CRC0 to CRCM, which would function in the same way. Therefore, the specific Conditional Read Circuits, CRC0 to CRCM, were chosen for illustrative purposes only and the invention is not limited to the specific embodiment

-26-

s shown. Consequently, the scope of the invention is
defined by the claims and their equivalents.

5